

Описание логической архитектуры платформы 1С:Предприятие

Оглавление

Обзор архитектуры платформы	2
Платформа и прикладные решения	2
Метаданные — способ описания прикладного решения.....	2
Построение прикладного решения на основе модели.....	3
Стандартные прототипы прикладных объектов.....	3
Согласованность технологий и инструментов	4
Многозвенная архитектура работы	4
Отказоустойчивый кластер с балансировкой нагрузки	5
Архитектура кластера.....	6
Основные возможности кластера серверов	6
Общая схема клиент-серверного варианта работы	6
Состав простейшего кластера серверов	8
Сеансы	9
Масштабируемость кластера	10
Использование нескольких менеджеров	10
Использование нескольких рабочих процессов.....	11
Использование нескольких рабочих серверов.....	11
Высокоуровневая модель интерфейса.....	11
Веб-клиент и тонкий клиент	12
Мобильная платформа	12
Интеллектуальные механизмы подготовки отчетов.....	13
Построение распределенных и интегрированных информационных систем.....	13
Облачные технологии	13
Технологический журнал.....	14
Журнал регистрации	15
Мониторинг состояния кластера	17
Механизмы аутентификации.....	19
Дата акселератор.....	19

Обзор архитектуры платформы

Основная задача платформы заключается в повышении уровня абстракции при разработке и использовании прикладных решений. Это позволяет перейти от технических и низкоуровневых понятий к более содержательным и высокоуровневым. Позволяет приблизить эти понятия к языку пользователей и специалистов в предметной области. В конечном итоге это значительно ускоряет и унифицирует разработку прикладного решения и его сопровождение.

Одновременно с этим платформа решает и традиционные задачи, связанные с производительностью, эргономикой, функциональностью и т. д.

Платформа и прикладные решения

В системе «1С:Предприятие 8» существует четкое разделение на платформу и прикладное решение. Платформа представляет собой framework, в котором функционирует прикладное решение:

- Платформа служит фундаментом для построения прикладных решений,
- Платформа является средой их исполнения,
- Платформа содержит инструментарий, необходимый для разработки, администрирования и поддержки прикладных решений.

При этом прикладное решение является самостоятельной сущностью и может выступать в качестве отдельного программного продукта. Но полностью опирается на технологии платформы.

Метаданные — способ описания прикладного решения

Прикладное решение не пишется в прямом смысле на языке программирования. Язык программирования используется только там, где это действительно необходимо.

В основе прикладного решения лежат метаданные. Они представляют собой структурированное декларативное его описание. Метаданные образуют иерархию объектов, из которых формируются все составные части прикладной системы и которые определяют все аспекты ее поведения. Фактически, при работе прикладного решения, платформа «проигрывает» (интерпретирует) метаданные, обеспечивая всю необходимую функциональность.

Метаданными описываются структуры данных, состав типов, связи между объектами, особенности их поведения и визуального представления, система разграничения прав доступа, пользовательский интерфейс и т. д. В метаданных сосредоточены сведения не только о том, «что хранить в базе данных», но и о том, «зачем» хранится та или иная информация, какова ее роль в системе, и как связаны между собой информационные массивы.

Использование языка программирования ограничено решением тех задач, которые действительно требуют алгоритмического описания, например, расчета налогов, проверки корректности введенных данных и т. д.

Построение прикладного решения на основе модели

В платформе заложена ориентация на построение прикладного решения на основе определенной модели. Под моделью понимается вся идеология построения прикладного решения. Сюда относятся способы построения структур данных, типы связей между данными, принципы манипулирования данными, формы описания бизнес-логики, способы связи данных с интерфейсными объектами, разделение функциональности по уровням системы и многое другое.

Важно, что все прикладные решения следуют принятой модели и этим обеспечивается единообразие и предсказуемость их поведения.

Стандартные прототипы прикладных объектов

В модели разработки «1С:Предприятия 8» все прикладное решение описывается метаданными в виде совокупности прикладных объектов, выбираемых из определенного набора прототипов (классов). Каждый такой прототип отвечает за отражение в прикладном решении определенной совокупности объектов или процессов предметной области, имеющих схожие поведенческие характеристики и сходную роль в общей картине решения.

Примерами таких прототипов являются «Справочники», «Документы», «Регистры накопления».

Каждый прототип имеет некоторую базовую реализацию, которая определяет особенности функционирования создаваемых на основе данного прототипа объектов: структуру хранимых сущностей вместе с некоторыми предопределенными полями, набор типов языка программирования, методы, свойства и события, а также типовые, для решаемой задачи, операции, способы отображения и редактирования, методы регулирования прав доступа и т. д.

Таким образом, все прикладное решение, фактически, состоит из объектов, четко разделенных по тем ролям, которые они играют в прикладном решении. Такой подход существенно усиливает эффект и от описания системы в терминах метаданных, и от построения приложения на основе модели.

Согласованность технологий и инструментов

Ключевым качеством платформы является достаточность ее средств для решения задач, стоящих перед прикладными решениями. Это позволяет обеспечить очень хорошую согласованность всех технологий и инструментов, которыми пользуется разработчик.

Простейший пример — система типов. В платформе разработчик использует одну систему типов данных и для взаимодействия с БД, и для реализации бизнес-логики, и для построения интерфейсных решений. Поэтому у него нет проблем, связанных с преобразованием типов при переходах между разными уровнями прикладной системы.

Другой пример это современная концепция работы с информацией, которая сочетает несколько способов представления данных:

- хранение сущностей в базе данных;
- их представление в языке программирования в виде объектов;
- их отображение в формате XML или JSON;
- представление структуры прикладного решения в модели REST (REST API).

Фактически любая информация может в зависимости от текущего режима работы представляться одним из этих способов. Долговременное хранение сущностей осуществляется в базе данных, что обеспечивает надежность и эффективную обработку больших объемов информации. Для внесения изменений данные предварительно преобразуются в объекты встроеного языка. При внутреннем обмене в распределенной базе данных, или взаимодействии с другими информационными системами, данные переносятся в формате XML или JSON. Также сторонние системы могут обращаться к прикладному решению с помощью HTTP запросов через REST интерфейс, автоматически генерируемый платформой. Ответ они могут получать в форматах Atom/XML или JSON.

Все эти способы представления опираются на единую систему понятий, и от разработчика не требуется усилий для трансформации данных из одного способа представления к другому. Он один раз описывает некоторую сущность, а разные представления данных платформа генерирует автоматически.

Многозвенная архитектура работы

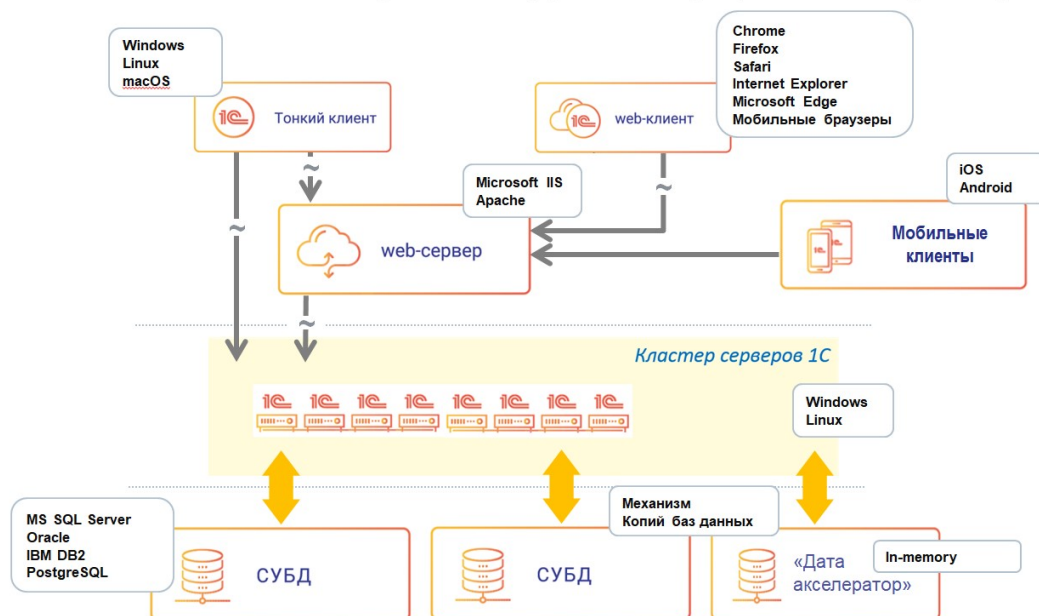
Прикладные решения, работающие под управлением платформы, используют многозвенную архитектуру «клиентское приложение — кластер серверов „1С:Предприятия 8“ — сервер базы данных».

С одной стороны это позволяет масштабировать систему от вариантов персонального использования, до работы в крупных, территориально распределенных холдинговых компаниях.

С другой стороны многозвенная архитектура позволяет выбирать между несколькими системами управления базами данных, которые будут использоваться для хранения прикладных данных.

Основные компоненты системы могут работать как под управлением операционной системы Windows, так и под управлением операционной системы Linux. Кроме этого клиентская часть «1С:Предприятия 8» может быть запущена и на компьютерах с операционными системами OS X и Apple iOS.

Логическая архитектура платформы 1С:Предприятие



Отказоустойчивый кластер с балансировкой нагрузки

Кластер серверов обеспечивает отказоустойчивость системы к таким событиям как выход из строя серверного компьютера (в том числе и центрального сервера), аварийное завершение рабочего процесса или менеджера кластера, физический разрыв соединения пользователя с кластером и последующее его восстановление. Это достигается благодаря тому, что существует возможность резервирования как самого кластера, так и рабочих процессов, функционирующих в нем. Кроме этого кластер предпринимает специальные действия для того, чтобы обеспечивать устойчивость к обрыву канала связи.

Кластер автоматически выполняет балансировку нагрузки между своими рабочими процессами на основе актуального анализа их доступной производительности. При этом администратор системы может корректировать нагрузку на отдельные рабочие серверы. Как за счет физического ограничения обслуживаемых информационных баз или соединений, так и за счет переназначения части функциональности кластера на другие рабочие серверы.

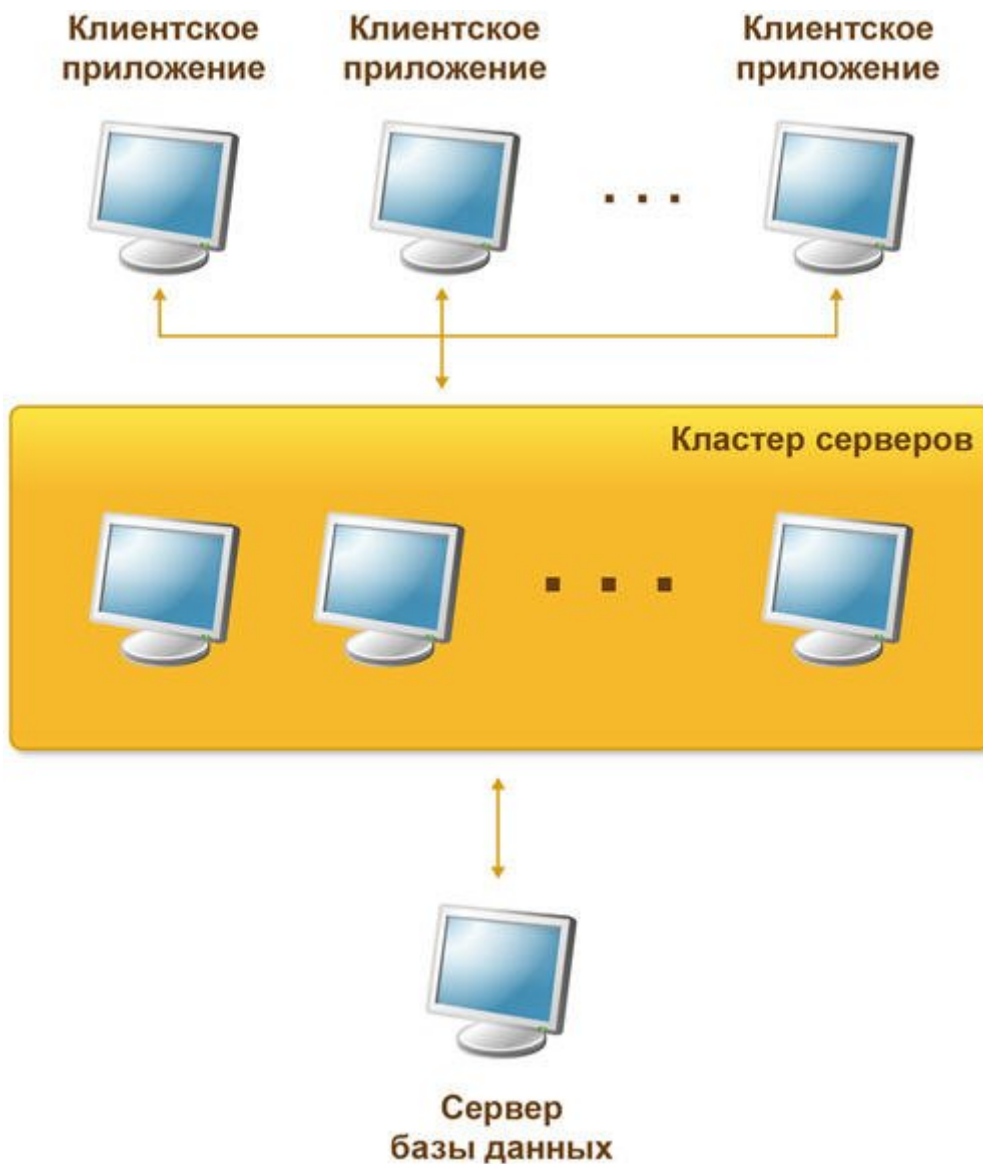
Архитектура кластера

Основные возможности кластера серверов

- может функционировать на одном или нескольких компьютерах (рабочих серверах);
- на каждом рабочем сервере может функционировать один или несколько рабочих процессов, обслуживающих клиентские соединения в рамках данного кластера;
- подключение новых клиентов к рабочим процессам кластера выполняется на основе анализа долгосрочной статистики загруженности рабочих процессов;
- взаимодействие процессов кластера с клиентскими приложениями, между собой и с сервером баз данных осуществляется по протоколу TCP/IP;
- процессы кластера сервера могут быть запущены как приложение, или как сервис.

Общая схема клиент-серверного варианта работы

В клиент-серверном варианте работы клиентское приложение взаимодействует с кластером серверов, который, в свою очередь, осуществляет взаимодействие с сервером баз данных.



Один из компьютеров, входящих в состав кластера серверов, является центральным сервером кластера. Центральный сервер, помимо обслуживания клиентских соединений, управляет работой всего кластера и хранит реестр кластера.

Для клиентского соединения кластер адресуется по имени центрального сервера и номеру сетевого порта. Если используется стандартный сетевой порт, то достаточно указания одного имени центрального сервера.

При установке соединения клиентское приложение обращается к центральному серверу кластера. Центральный сервер, на основе анализа статистики загруженности рабочих процессов, направляет клиентское приложение к конкретному рабочему процессу, который будет его обслуживать. Этот процесс может находиться как на центральном сервере, так и на любом рабочем сервере кластера.

Рабочий процесс выполняет аутентификацию пользователя и обслуживает соединение до окончания сеанса работы клиента с данной информационной базой.

Состав простейшего кластера серверов

Простейший кластер серверов может располагаться на одном компьютере и содержать один рабочий процесс:



На рисунке представлены все элементы, которые задействованы в работе кластера серверов, а именно:

- процессы кластера серверов:
 - ragent.exe;
 - rnmgr.exe;
 - rphost.exe;
- хранилища данных:
 - список кластеров;
 - реестр кластера.

Функционирование компьютера в составе кластера обеспечивается процессом ragent.exe, который называется агентом сервера. Соответственно компьютер, на котором запущен агент сервера, называется рабочим сервером. Одной из функций агента сервера является ведение списка кластеров, расположенных на данном рабочем сервере.

Агент сервера и список кластеров не входят в состав кластера серверов, а лишь обеспечивают работу сервера и кластеров, которые расположены на нем.

Непосредственно кластер серверов включает в себя следующие элементы:

- один или несколько процессов rnmgr.exe;
- реестр кластера;
- один или несколько процессов rphost.exe.

Процесс rnmgr.exe называется менеджером кластера. Этот процесс управляет функционированием всего кластера. В составе кластера может существовать несколько процессов менеджера кластера. Один из этих процессов всегда является главным менеджером кластера, остальные — дополнительными менеджерами. Рабочий сервер, на котором функционирует главный менеджер кластера и располагается реестр кластера, называется центральным сервером кластера. Одной из функций главного менеджера кластера является ведение реестра кластера.

Процесс rphost.exe называется рабочим процессом. Рабочий процесс обслуживает непосредственно клиентские приложения, взаимодействует с сервером баз данных и в нем, в частности, могут исполняться процедуры серверных модулей конфигурации.

Сеансы

Сеанс определяет активного пользователя информационной базы и поток управления этого пользователя. Существуют следующие виды сеансов:

- Толстый клиент, Тонкий клиент, Веб-клиент — создаются при обращении, соответственно, толстого, тонкого и веб-клиента к информационной базе,
- Конфигуратор — создается при обращении конфигуратора к информационной базе,
- СОМ-соединение — создается при обращении к информационной базе через внешнее соединение,

- WS-соединение — создается при обращении веб-сервера к информационной базе в результате обращения к Web-сервису, опубликованному на веб-сервере,
- Фоновое задание — создается при обращении рабочего процесса кластера к информационной базе. Предназначен для выполнения кода процедуры фонового задания,
- Консоль кластера — создается при обращении утилиты администрирования клиент-серверного варианта к рабочему процессу,
- СОМ-администратор — создается при обращении к рабочему процессу через внешнее соединение.

Масштабируемость кластера

Масштабируемость кластера серверов может осуществляться несколькими способами:

- за счет увеличения количества менеджеров кластера и распределения между ними сервисов;
- за счет увеличения количества рабочих процессов, функционирующих на конкретном рабочем сервере;
- за счет увеличения количества рабочих серверов, входящих в состав кластера.

Использование нескольких менеджеров

Все функции, выполняемые менеджером кластера, разделены на несколько сервисов. Эти сервисы могут быть назначены различным менеджерам кластера, что позволяет распределить всю нагрузку между несколькими процессами.

Часть сервисов может использоваться только на главном менеджере кластера:

- сервис конфигурации кластера,
- сервис блокировок кластера,
- сервис управления предметами отладки.

Остальные сервисы могут быть назначены произвольным менеджерам кластера:

- сервис журналов регистрации,
- сервис полнотекстового поиска,
- сервис заданий,
- сервис нумерации,
- сервис пользовательских настроек,
- сервис времени,
- сервис блокировки объектов,
- сервис сеансовых данных,

- сервис транзакционных блокировок.

Использование нескольких рабочих процессов

Использование нескольких рабочих процессов, с одной стороны, позволяет снизить нагрузку на каждый конкретный рабочий процесс. С другой стороны, запуск нескольких рабочих процессов позволяет более эффективно использовать аппаратные ресурсы рабочего сервера. Кроме этого запуск нескольких рабочих процессов позволяет повысить надежность сервера, изолировав группы клиентов, работающих с разными информационными базами. В кластере, в котором возможен запуск нескольких рабочих процессов, рабочий процесс может быть автоматически перезапущен по истечении времени, установленного администратором кластера.

Использование нескольких рабочих серверов

Увеличение количества рабочих серверов, входящих в кластер, позволяет использовать большее количество рабочих процессов (обслуживать большее количество клиентских соединений), не увеличивая при этом нагрузку на каждый конкретный рабочий процесс.

Высокоуровневая модель интерфейса

Основной идеей построения интерфейса является максимальное использование информации из метаданных, а также объектов манипулирования данными с тем, чтобы вся конструкция не требовала детальной настройки со стороны разработчика и функционировала по большей части автоматически.

Разработчику достаточно связать такой объект с элементом формы или с самой формой, и механизм интерфейса полностью возьмет на себя организацию просмотра и модификации данных. Платформа автоматически подключит расширения, учитывающие тип данных, с которыми связан элемент управления или форма.

Платформа содержит целый набор механизмов, позволяющих создавать приложения на разных языках. Начиная от различных языков интерфейса платформы, поддержки национальных дат, чисел, и заканчивая средствами редактирования текстов интерфейса.

Веб-клиент и тонкий клиент

Для работы с прикладным решением пользователь может выбрать одно из двух клиентских приложений: тонкий клиент, либо веб-клиент. Каждое из клиентских приложений обладает своими преимуществами.

Тонкий клиент более функциональный за счёт того, что устанавливается на компьютере пользователя. Например, он имеет непосредственный доступ к файловой системе клиентского компьютера, может использовать локальные лицензии, установленные на компьютере пользователя.

В то же время веб-клиент более универсальный за счёт того, что не требует предварительной установки. Он исполняется не в среде операционной системы компьютера, а в среде интернет-браузера. Поэтому пользователю достаточно всего лишь запустить свой браузер, ввести адрес веб-сервера, на котором опубликована информационная база — и веб-клиент «сам придет» к нему на компьютер и начнет исполняться. С помощью веб-клиента можно работать с прикладными решениями на компьютерах, которые заранее не подготовлены (или не могут быть подготовлены) для этого.

Важным фактом является то, что разработка прикладного решения ведется независимо от того, какое из клиентских приложений будет использоваться для работы с приложением. Клиентские модули, разработанные в конфигурации, веб-клиент автоматически компилирует из встроенного языка «1С:Предприятия 8» и непосредственно исполняет на своей стороне.

Мобильная платформа

Для работы на смартфонах и планшетах предоставляется специальная технология — мобильная платформа. С её помощью создаются приложения, которые автономно функционируют на этих устройствах. Основное назначение таких приложений заключается в том, чтобы быть частью распределенной информационной системы, обмениваться данными с основным приложением. А в те периоды, когда связь с основным приложением невозможна, обеспечивать полноценную автономную работу.

Разработка и отладка мобильных приложений ведётся теми же инструментами, в тех же терминах и понятиях, что и разработка обычных, «настольных» приложений. Отличие заключается лишь в том, что в конце, для получения дистрибутива мобильного приложения нужно выполнить ряд дополнительных действий, «собрать» мобильное приложение.

Интеллектуальные механизмы подготовки отчетов

Средства подготовки отчетности тесно интегрированы с другими механизмами платформы и имеют мощные возможности для интерактивной работы. Благодаря этому отчеты органично вписываются в общий интерфейс приложения. Фактически, пользователь в процессе работы не видит грани между общим интерфейсом и механизмом отчетности.

Одним из наиболее интересных механизмов такого рода является система компоновки данных, предоставляющая возможность с минимальными усилиями получить отчет с развитой функциональностью.

Построение распределенных и интегрированных информационных систем

Платформа содержит мощный набор механизмов обмена, способный решать самые разнообразные задачи. От поддержки территориально распределенных информационных баз, до построения сложных гетерогенных информационных систем, включающих, наряду с решениями на платформе «1С:Предприятие 8», еще и внешние приложения.

Кроме этого в платформе имеется целый набор средств, позволяющих интегрироваться практически с любыми внешними программами и оборудованием на основе общепризнанных открытых стандартов и протоколов передачи данных. Это и технологии веб-сервисов (HTTP, REST, SOAP), и различные форматы обмена данными (JSON, XML), различные протоколы (HTTP, FTP) и т.д.

Облачные технологии

Прикладные решения «1С:Предприятия 8» способны функционировать в облаке, благодаря набору технологий и механизмов.

Базисом для облачных технологий является механизм разделения данных, реализованный в платформе. Благодаря ему прикладные решения могут работать в архитектуре multitenancy, когда единый экземпляр объекта приложения, запущенного на сервере, обслуживает множество клиентов или организаций.

Другим важным элементом является отказоустойчивый масштабируемый кластер серверов, обслуживающий большое количество одновременно работающих клиентов. Работа в облаке предполагает также подключение к информационной базе по протоколу HTTP (HTTPS), благодаря чему клиенты могут работать через интернет из любой точки земного шара.

Завершающим элементом облачных технологий является инфраструктура сервиса, позволяющая развертывать приложения «1С:Предприятия 8» в модели SaaS. Эта бизнес-модель подразумевает, что поставщик разрабатывает прикладное решение и самостоятельно управляет им, предоставляя потребителю доступ к прикладному решению через Интернет. Такой подход избавляет потребителя от всех затрат, связанных с установкой, обновлением и поддержкой оборудования и программного обеспечения. Потребитель оплачивает лишь пользование услугой.

Технологический журнал

Технологический журнал системы 1С:Предприятие 8 может использоваться для анализа технологических проблем работы системы и анализа аварийных завершений. Он регистрирует информацию от всех приложений системы 1С:Предприятие 8, работающих на данном компьютере.

Технологический журнал представляет собой совокупность текстовых файлов, хранящихся в указанном каталоге.

Файлы технологического журнала могут быть просмотрены с помощью любого текстового редактора.

По умолчанию технологический журнал включен, и настроен на сохранение минимальных дампов при аварийном завершении приложения в каталог %USERPROFILE%\Local Settings\Application Data\1C\1Cv8\dumps. При необходимости может быть выполнена произвольная настройка журнала регистрации с помощью конфигурационного файла logcfg.xml.

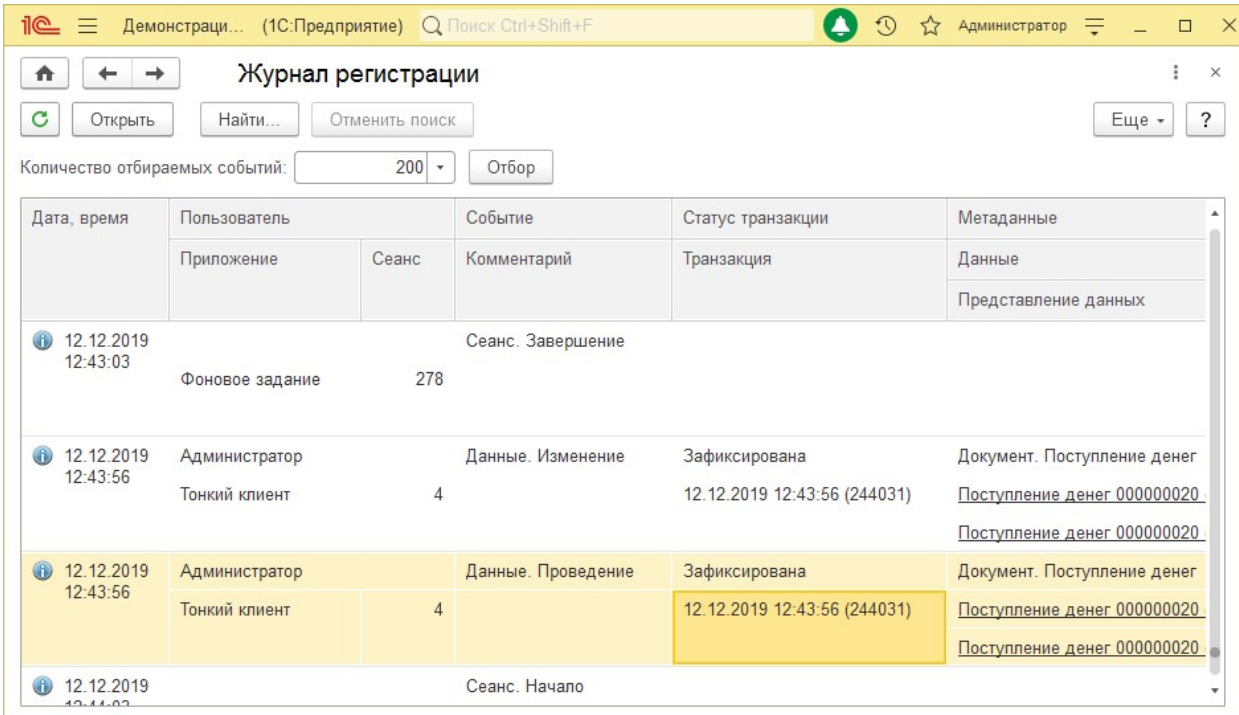
Ниже приведен пример простейшего конфигурационного файла, который указывает что:

- в технологическом журнале регистрируются все возможные события;
- файлы технологического журнала располагаются в каталоге d:\v81\logs;
- файлы технологического журнала хранятся в течение одного часа;
- файлы дампа помещаются в каталог d:\v81\dumps;
- файлы дампа содержат всю доступную информацию (содержимое всей памяти процесса).

```
- <config xmlns="http://v8.1c.ru/v8/tech-log">
- <log history="1" location="d:\v81\logs">
- <event>
- <ne property="Name" value="" />
- </event>
- <property name="all" />
- </log>
- <dump create="1" location="d\v81\dumps" type="2" />
- </config>
```

Журнал регистрации

Журнал регистрации содержит информацию о том, какие события происходили в информационной базе в определенный момент времени или какие действия выполнял тот или иной пользователь. Для каждой записи журнала, отражающей изменение данных, отображается статус завершения транзакции (транзакция завершена успешно, или же транзакция отменена). Это позволяет понять изменены реально данные или нет:



Дата, время	Пользователь		Событие	Статус транзакции	Метаданные
	Приложение	Сеанс	Комментарий	Транзакция	Данные Представление данных
12.12.2019 12:43:03	Фоновое задание	278	Сеанс. Завершение		
12.12.2019 12:43:56	Администратор Тонкий клиент	4	Данные. Изменение	Зафиксирована 12.12.2019 12:43:56 (244031)	Документ. Поступление денег Поступление денег 000000020 Поступление денег 000000020
12.12.2019 12:43:56	Администратор Тонкий клиент	4	Данные. Проведение	Зафиксирована 12.12.2019 12:43:56 (244031)	Документ. Поступление денег Поступление денег 000000020 Поступление денег 000000020
12.12.2019 12:44:03			Сеанс. Начало		

Для событий успешной и неуспешной аутентификации в информационной базе 1С:Предприятия в журнал записывается, какой именно пользователь операционной системы выполняет эту аутентификацию.

Для событий доступа к данным и отказа в доступе к данным можно гибко настроить состав регистрируемой информации. Набор полей объектов конфигурации, при доступе к которым будет регистрироваться событие, и состав дополнительной информации, которая будет записываться в журнал регистрации при наступлении этого события. Например, можно указать, что в журнал будут заноситься записи о том, что пользователь прочитал сумму начислений из регистра, хранящего данные о заработной плате. При этом в журнал будет занесена информация не только о том пользователе, который прочитал эти данные, но и информация о том, начисления какому именно сотруднику были прочитаны.

Журнал регистрации доступен как в режиме 1С:Предприятие, так и в режиме Конфигуратор.

В режиме 1С:Предприятие по щелчку мыши в полях Данные и Представление данных можно перейти к тому объекту прикладного решения, который указан в записи журнала регистрации.

Информацию, находящуюся в журнале регистрации, можно отбирать по большому количеству критериев. Например, можно отобразить только информацию о том, какие документы изменялись определенным пользователем в заданный промежуток времени:

Настройка отбора журнала регистрации событий

Интервал отображения, с: [] по: []

Основные свойства

Пользователи: []
События: []
Компьютеры: []
Приложения: []
Комментарий: []

Важность:

- Ошибка
- Предупреждение
- Информация
- Примечание

Данные

Метаданные: []
Данные: []
Представление данных: []

Прочие

Статус транзакции:

- Нет транзакции
- Зафиксирована
- Не завершена
- Отменена

Транзакция: []

Сеансы: []
Рабочие серверы: []
Основные IP порты: []
Вспомогательные IP порты: []

OK Отмена ?

Существует возможность настройки уровня событий, отображаемых в журнале регистрации, а также периодичности разделения журнала на отдельные файлы (включение/выключение журнала регистрации возможно как интерактивно, так и средствами встроенного языка):

Настройка журнала регистрации

Регистрировать в журнале события

- Не регистрировать
- Регистрировать ошибки
- Регистрировать ошибки, предупреждения
- Регистрировать ошибки, предупреждения, информацию
- Регистрировать ошибки, предупреждения, информацию, примечания

Изменить формат

OK
Отмена
Справка
Сократить

Кроме этого разработчик может самостоятельно добавлять записи в журнал регистрации, используя средства встроенного языка. Такая возможность позволяет настраивать журнал регистрации под нужды конкретного прикладного решения.

Используя средства работы со списками, разработчик имеет возможность выгрузить журнал регистрации в текстовый или табличный документ, который в дальнейшем может быть сохранен в собственном формате или, например, формате листа Excel или документа HTML.

Кроме этого поддерживается выгрузка журнала регистрации в формате XML, что может использоваться для последующего анализа выгруженных записей средствами встроенного языка.

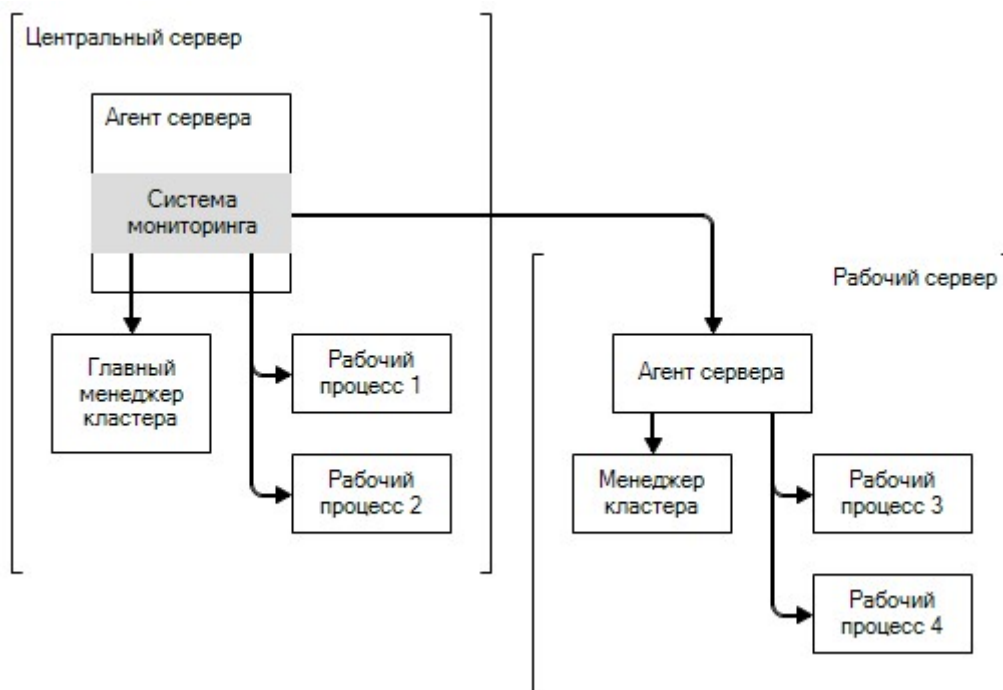
Особенности функционирования журнала регистрации

- при создании новой информационной базы для журнала устанавливается режим регистрации событий всех уровней важности;
- при загрузке информационной базы из файла журнал регистрации не очищается.

Мониторинг состояния кластера

Система мониторинга это комплексное решение, позволяющее повысить защищённость сервера «1С:Предприятия 8» от ошибок, которые могут возникнуть в его рабочих процессах. Это могут быть самые разные ошибки. Они могут быть следствием некорректной работы платформы. Или они могут возникнуть в результате выполнения некорректного прикладного кода, который исполняют рабочие процессы сервера. Задача системы мониторинга заключается в том, чтобы своевременно обнаружить проблему и автоматически её исправить.

Система мониторинга внедрена в процесс агента сервера. Каждые 10 секунд она производит опрос процессов кластера. В кластер может объединяться несколько рабочих серверов, каждым из которых управляет собственный агент сервера. Поэтому опрос процессов кластера производит только агент, управляющий центральным сервером:



Опрашиваются все процессы, функционирующие в кластере: менеджеры кластера, рабочие процессы. Процессы, исполняемые на рабочих серверах, опрашиваются через агентов этих серверов. Таким образом дополнительно проверяется ещё и работоспособность самих агентов.

Каждый процесс система мониторинга проверяет по следующим критериям:

- соединение с процессом; оно должно быть установлено в течение 20 секунд;
- стандартный запрос (тест скорости выполнения, соединения с базой данных, дисковые операции);
- объем памяти, занимаемой процессом;
- количество ошибок на количество запросов;
- завершение процессов, удаленных из реестра кластера; такие процессы должны завершиться в течение 20 минут.
- Результаты проверки записываются в технологический журнал. Если система мониторинга обнаруживает проблемный процесс, то она может завершить его самостоятельно, создавая перед этим дампы памяти процесса.

Опциями системы мониторинга можно управлять интерактивно, с помощью утилиты администрирования клиент-серверного варианта, а также программно с помощью встроенного языка или с помощью кроссплатформенного интерфейса администрирования кластера.

Механизмы аутентификации

Механизм аутентификации — это один из инструментов администрирования. Он позволяет определить, кто именно из пользователей, перечисленных в списке пользователей системы, подключается к прикладному решению в данный момент.

Система поддерживает несколько видов аутентификации, которые могут использоваться в зависимости от конкретных задач, стоящих перед администратором информационной базы:

- OpenID Connect аутентификация
- OpenID аутентификация
- Аутентификация 1С:Предприятия
- Аутентификация операционной системы
- Двухфакторная аутентификация

Если для пользователя не указан ни один из видов аутентификации, — такому пользователю доступ к прикладному решению закрыт.

Дата акселератор

Дата акселератор — собственная разработка фирмы «1С», предназначенная для ускорения работы с большими объемами данных для аналитических отчетов. Технология представляет собой специальную систему управления базой данных (СУБД), в которой данные целиком размещаются для обработки в оперативной памяти сервера (in-memory DB) и хранятся в структурах, оптимизированных для работы аналитических отчетов. Дата акселератор предоставляется в качестве сервиса в составе кластера серверов «1С:Предприятия» и работает только совместно с механизмом копий баз данных.

Дата акселератор оптимизирован для выполнения аналитических запросов, обрабатывающих большие объемы данных и возвращающих в качестве результата относительно небольшое количество записей. Это является рекомендованным сценарием использования механизма. Для построения систем оперативного анализа данных класса Business Intelligence рекомендуется использование связки продуктов Дата акселератор и 1С:Аналитика.

Для одной информационной базы можно использовать несколько дата акселераторов и распределять нагрузку между ними. Например, можно случайным образом направлять запросы пользователя на один из имеющихся дата акселераторов или разбивать отчеты на группы, указывая, на каком акселераторе будет выполняться каждая группа отчетов.